

IN THE CLAIMS:

1. (Currently amended) A method of debugging a software program whose execution flow is responsive at least in part to a first set of options, ~~said software program executing properly with at least one of a first set of options and executing improperly with at least one of a second set of options~~, said method comprising the steps of:

generating a first log file by executing said program with ~~said at least one of said~~ first set of options, said first log file including an indication of all functions executed by said program during this first execution;

modifying at least one of the first set of options, and then generating a second log file by executing said program with ~~said at least one of said second~~ modified first set of options, said second log file including an indication of all functions executed by said program during this second execution; and

comparing said first log file with said second log file to debug the software program.

2. (Original) The method according to claim 1, further comprising the steps of:

generating said first log file including a first set of return codes;

generating said second log file including a second set of return codes; and

comparing said first set of return codes with said second set of return codes to debug the software program.

3. (Currently amended) The method according to claim 1, further comprising the ~~step~~ steps of:

compiling said software program to generate compiled code, said compiled code including a listing of said functions;

generating a file including said listing obtained from said compiled code; and

generating a debug script utilizing said file.

4. (Original) The method according to claim 3, further comprising the step of compiling said software program utilizing a C compiler and utilizing a “-g” option, said “-g” option generating said listing of said function.
5. (Canceled)
6. (Currently amended) The method according to claim 5 ~~3~~, further comprising the step of utilizing a UNIX dump command to generate said file, said UNIX dump command causing an output of said listing.
7. (Canceled)
8. (Original) The method according to claim 1, further comprising the steps of:
automatically generating a debug script including the steps of:
generating script code for each of a plurality of function calls included in said software program, said script code setting a breakpoint at each of said plurality of function calls;
generating script code which logs each of a plurality of said plurality of functions calls executed by said software program when said software program is executed under the control of said debug program; and
generating script code which causes execution of said software program to continue after each of said plurality of said plurality of function calls is logged.
9. (Currently amended) A computer program product for debugging a software program whose execution flow is responsive at least in part to a first set of options, ~~said software program executing properly with at least one of a first set of options and executing improperly with at least one of a second set of options~~, said computer program product comprising:
instruction means for generating a first log file by executing said program with ~~said at least one of~~ said first set of options, said first log file including an indication of all functions executed by said program during this first execution;

instruction means for modifying at least one of the first set of options, and then generating a second log file by executing said program with said ~~at least one of said~~ second modified first set of options, said second log file including an indication of all functions executed by said program during this second execution; and

instruction means for comparing said first log file with said second log file to debug the software program.

10. (Original) The product according to claim 9, further comprising:

instruction means for generating said first log file including a first set of return codes;

instruction means for generating said second log file including a second set of return codes; and

instruction means for comparing said first set of return codes with said second set of return codes to debug the software program.

11. (Currently amended) The product according to claim 9, further comprising:

instruction means for compiling said software program to generate compiled code, said compiled code including a listing of said functions;

instruction means for generating a file including said listing obtained from said compiled code; and

instruction means for generating a debug script utilizing said file.

12. (Original) The product according to claim 11, further comprising instruction means for compiling said software program utilizing a C compiler and utilizing a “-g” option, said “-g” option generating said listing of said function.

13. (Canceled)

14. (Currently amended) The product according to claim ~~13~~ 11, further comprising instruction means for utilizing a UNIX dump command to generate said file, said UNIX dump command causing an output of said listing.

15. (Canceled)

16. (Original) The product according to claim 9, further comprising:

instruction means for automatically generating a debug script including:

instruction means for generating script code for each of a plurality of function calls included in said software program, said script code setting a breakpoint at each of said plurality of function calls;

instruction means for generating script code which logs each of a plurality of said plurality of functions calls executed by said software program when said software program is executed under the control of said debug program; and

instruction means for generating script code which causes execution of said software program to continue after each of said plurality of said plurality of function calls is logged.

17. (Currently amended) A system for debugging a software program whose execution flow is responsive at least in part to a first set of options, ~~said software program executing properly with at least one of a first set of options and executing improperly with at least one of a second set of options~~, comprising:

a first log file being generated by executing said program with ~~said at least one of~~ said first set of options, said first log file including an indication of all functions executed by said program during this first execution;

a second log file being generated by executing said program with ~~said at least one of said second~~ a modified first set of options, said second log file including an indication of all functions executed by said program during this second execution; and

means for comparing said first log file with said second log file to debug the software program.

18. (Original) The system according to claim 17, further comprising:

said first log file being generated including a first set of return codes;

said second log file being generated including a second set of return codes; and

means for comparing said first set of return codes with said second set of return codes to debug the software program.

19. (Currently amended) The system according to claim 17, further comprising:
said software program being compiled to generate compiled code, said compiled code including a listing of said functions;

a file being generated including said listing obtained from said compiled code;
and

a debug script being generated utilizing said file.

20. (Original) The system according to claim 19, further comprising said software program being compiled utilizing a C compiler and utilizing a “-g” option, said “-g” option generating said listing of said function.

21. (Canceled)

22. (Currently amended) The system according to claim ~~24~~ 19, further comprising a UNIX dump command being utilized to generate said file, said UNIX dump command causing an output of said listing.

23. (Canceled)

24. (Original) The system according to claim 17, further comprising:

a debug script being automatically generated including:

script code being generated for each of a plurality of function calls included in said software program, said script code setting a breakpoint at each of said plurality of function calls;

script code being generated which logs each of a plurality of said plurality of functions calls executed by said software program when said software program is executed under the control of said debug program; and

script code being generated which causes execution of said software program to continue after each of said plurality of said plurality of function calls is logged.

25. (New) A method of debugging a software program, said method comprising the steps of:

analyzing said software program to determine a listing of executable functions in the software program;

generating a file including said listing of executable functions; and

generating a debug script for a debug program utilizing said file.

26. (New) The method of Claim 25, wherein the step of generating a debug script comprises:

utilizing said file, including said listing of executable functions, to generate script code which logs executable functions executed by said software program when said software program is executed under the control of said debug program.

27. (New) The method of Claim 26, further comprising:

setting at least one parameter used by the software program;

executing said software program using said debug script to generate a first log file;

changing the at least parameter and then re-executing said software program using said debug script to generate a second log file; and

comparing said first log file and said second log file to debug the software program.

28. (New) The method according to Claim 25, wherein the analyzing step is done by a compiler, and further comprising the step of utilizing a dump command to generate said file, said dump command causing an output of said listing.

29. (New) A computer program product for debugging a software program, said computer program product comprising:

instruction means for analyzing said software program to determine a listing of executable functions in the software program;

instruction means for generating a file including said listing of executable functions; and

instruction means for generating a debug script for a debug program utilizing said file.

30. (New) The computer program product of Claim 29, wherein the instruction means for generating a debug script comprises:

instruction means for utilizing said file, including said listing of executable functions, to generate script code which logs executable functions executed by said software program when said software program is executed under the control of said debug program.

31. (New) The computer program product of Claim 30, further comprising:

instruction means for setting at least one parameter used by the software program;

instruction means for executing said software program using said debug script to generate a first log file;

instruction means for changing the at least parameter and then re-executing said software program using said debug script to generate a second log file; and

instruction means for comparing said first log file and said second log file to debug the software program.

32. (New) The computer program product according to Claim 29, wherein the instruction means for analyzing said software program is a compiler, and further comprising a dump command means for generating said file, said dump command means causing an output of said listing.